

降本增效下 Kafka 的稳定性实践

腾讯云 / LoboXu(许文强)

目录

1. 降本增效背景下大规模 **Kafka** 集群面临的稳定性挑战
2. 如何通过弹性的计算存储架构来提高集群的稳定性
3. 混沌工程在云上大规模 **kafka** 集群的落地实践
4. 云上消息队列在降本、稳定性、架构升级之间权衡的一些思考

自我介绍

腾讯云高级工程师，腾讯云消息队列 **Kafka** 技术负责人，专注于中间件、消息队列、**Serverless**领域。

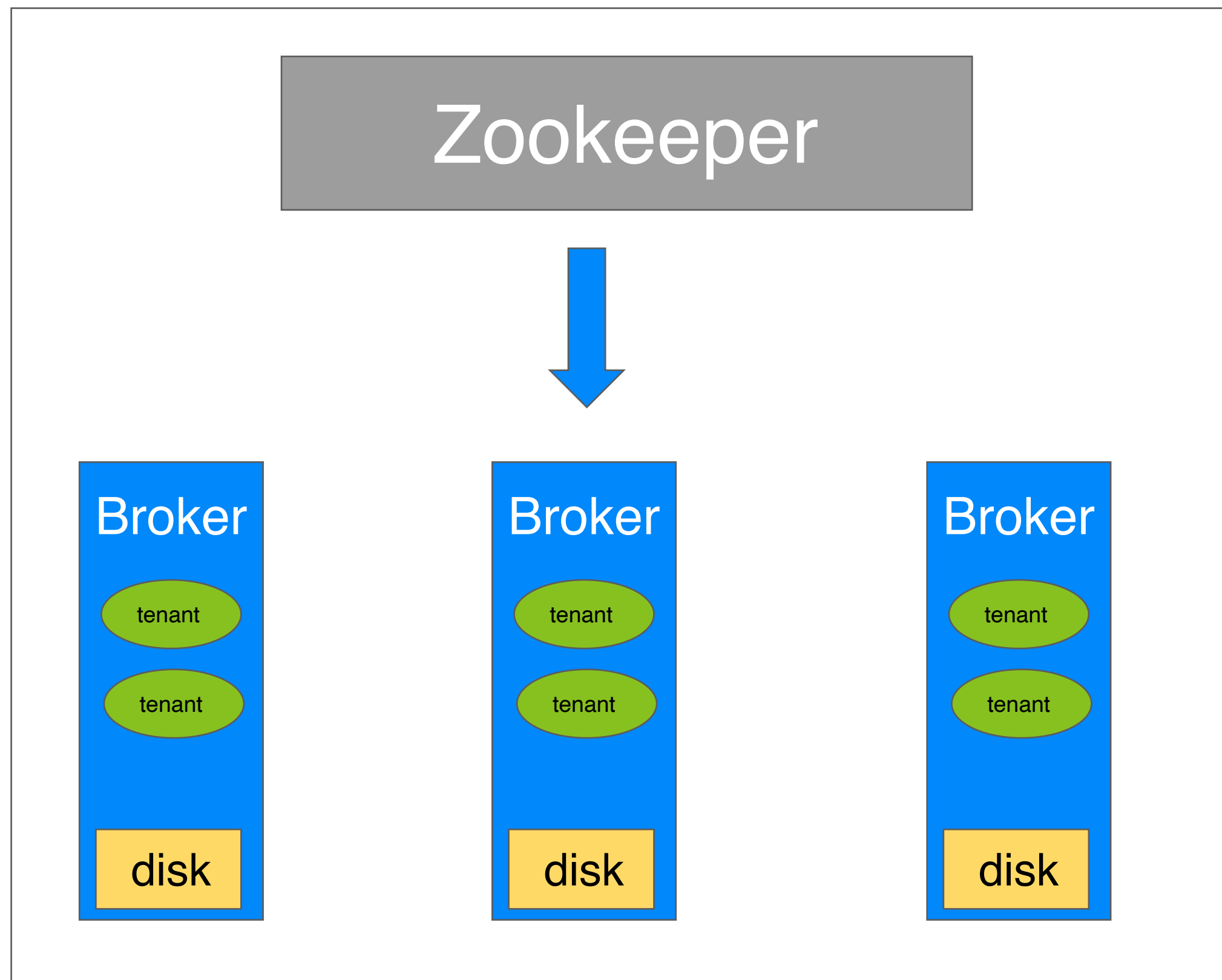
- 极客时间《消息队列核心技术40讲》出品人
- 技术书籍《基础软件之路：企业级实践与开源战略》联合作者。
- **Apache Kafka/RocketMQ/Pulsar Contributor**

对消息队列在云上的商业化建设、技术架构设计、开源代码升级、运营运维体系建设具有丰富的经验。



降本增效背景下 大规模Kafka集群面临的稳定性挑战

公有云Kakfa的部署和成本结构



基于 Zookeeper 构建的集群

1. 基于 Zookeeper 和Kraft 构建集群
2. 和开源自建架构、成本结构基本一致

节点、硬盘、带宽、其他



提升利用率

按需交付

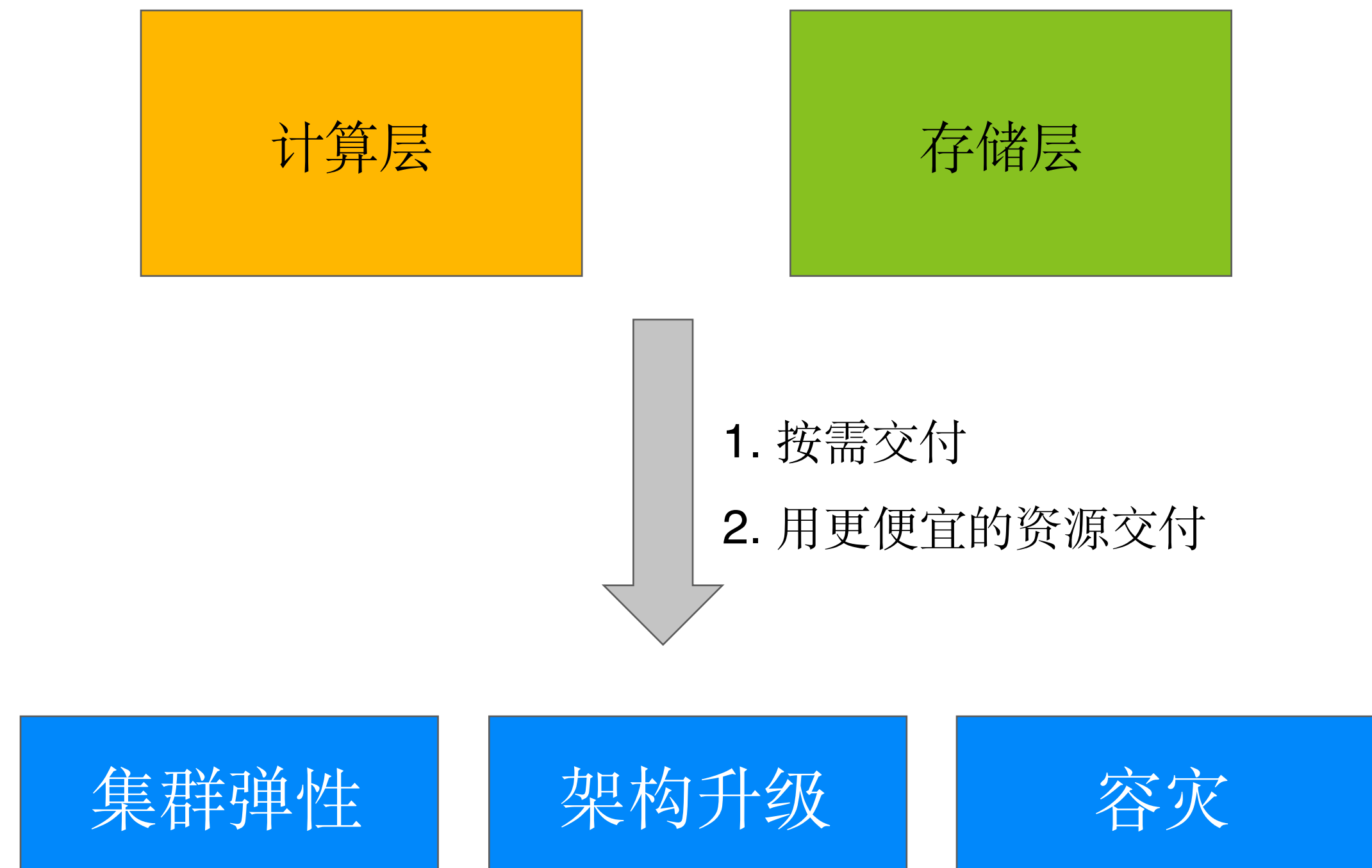
弹性 Serverless

1. 实际的利用率不高
2. 不同时间段波峰波谷明显

成本优化后Kakfa面临的稳定性挑战

Kafka 常见稳定性问题:

1. 客户端超时
2. 生产写入失败
3. 生产写入慢
4. 消费堆积
5. 硬盘打满
6. CPU打满 & GC严重
7. 节点、机房、可用区故障



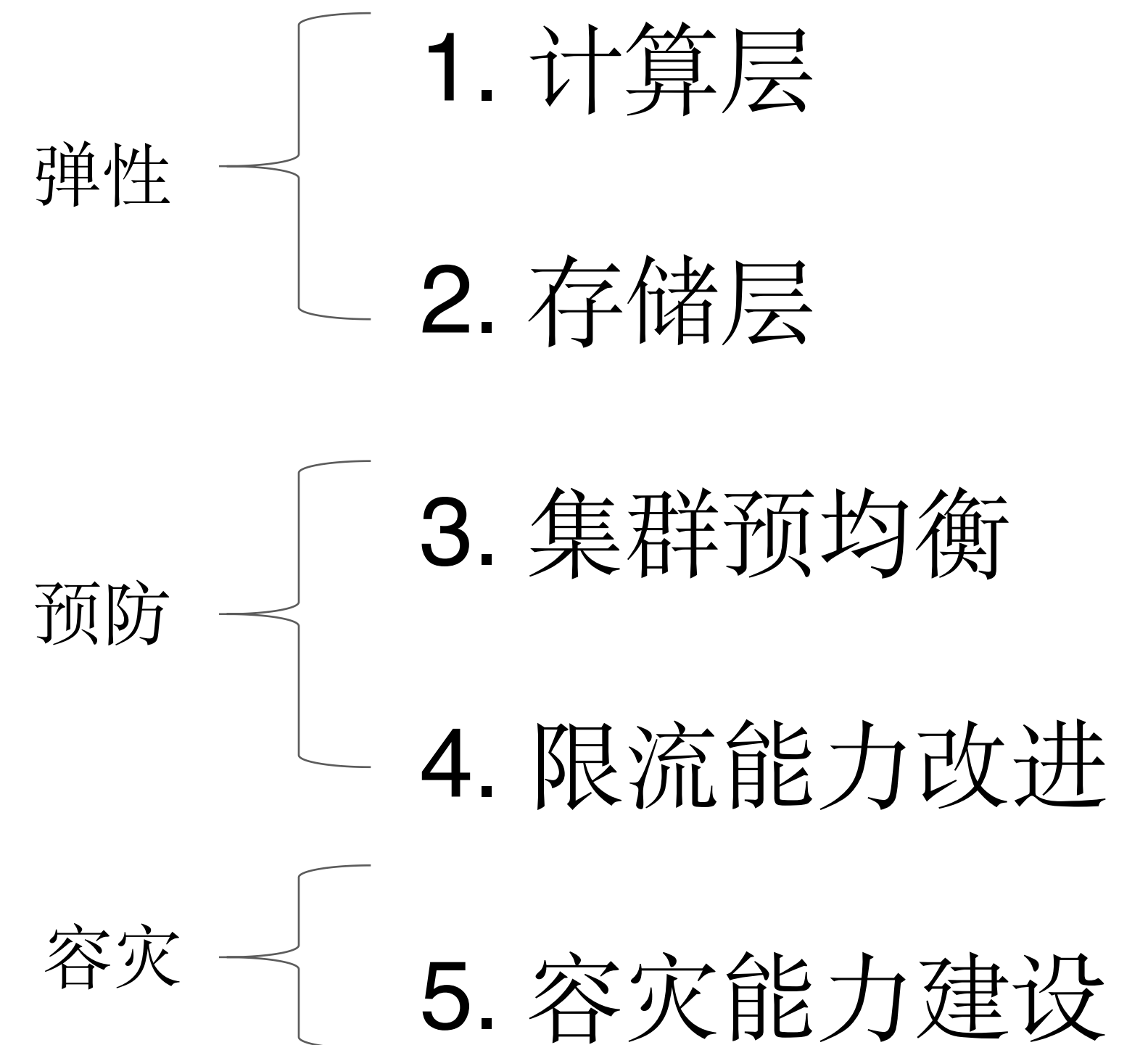
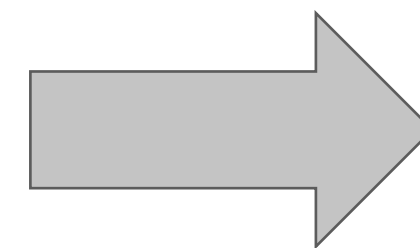
如何通过弹性的计算存储架构 来提高集群的稳定性

分享三个典型现网案例

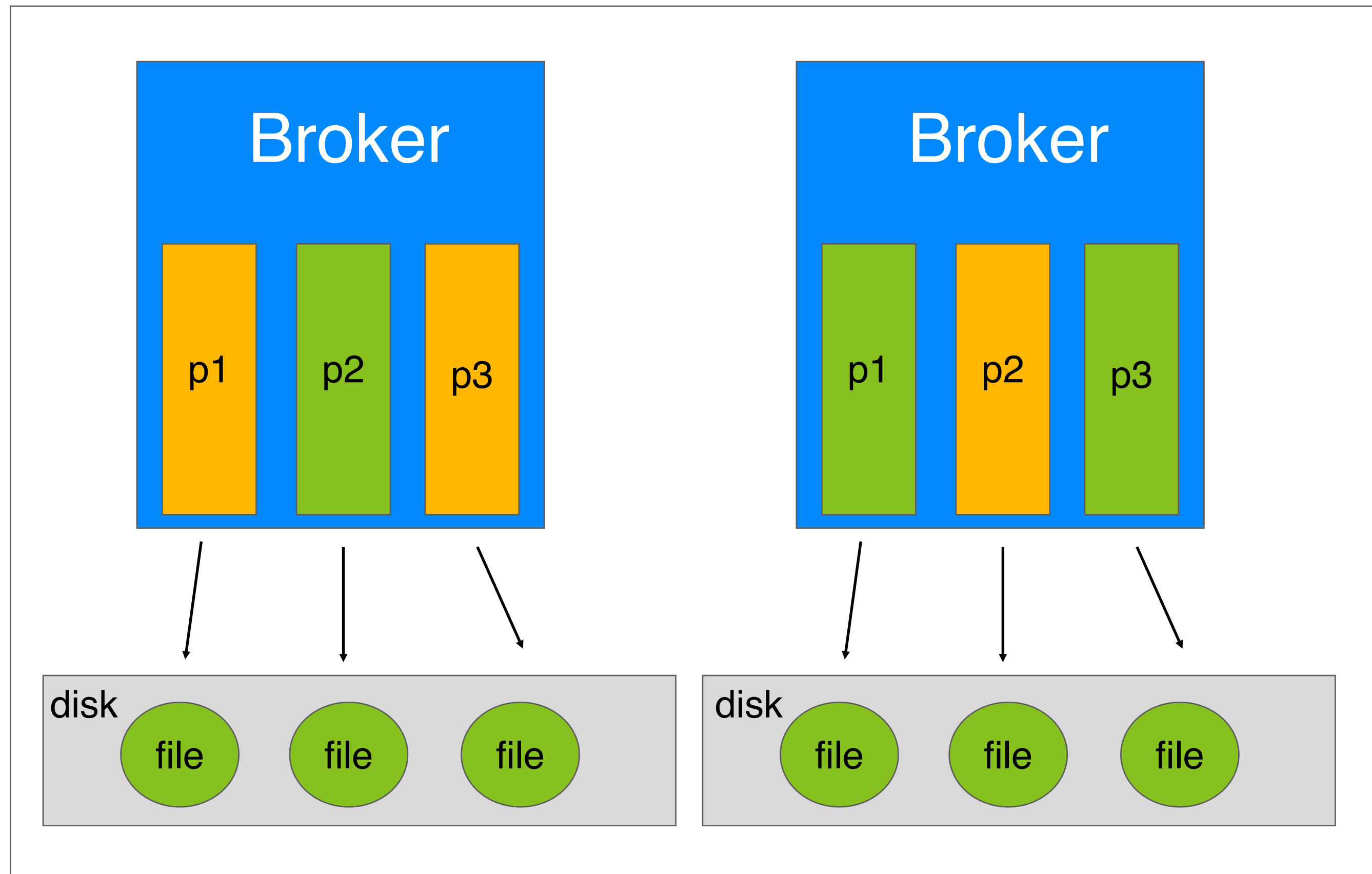
1. 共享集群的A用户的流量上升导致某个节点负载高，从而导致B客户的生产耗时从几十ms提高到了500~1000ms。

2. 经过分析业务的流量趋势和历史使用情况后，我们进行了硬盘缩容操作。在某一天，用户的业务流量快速上涨，把硬盘打满，从而影响了读写。

3. 某可用区故障后，影响了客户的业务。



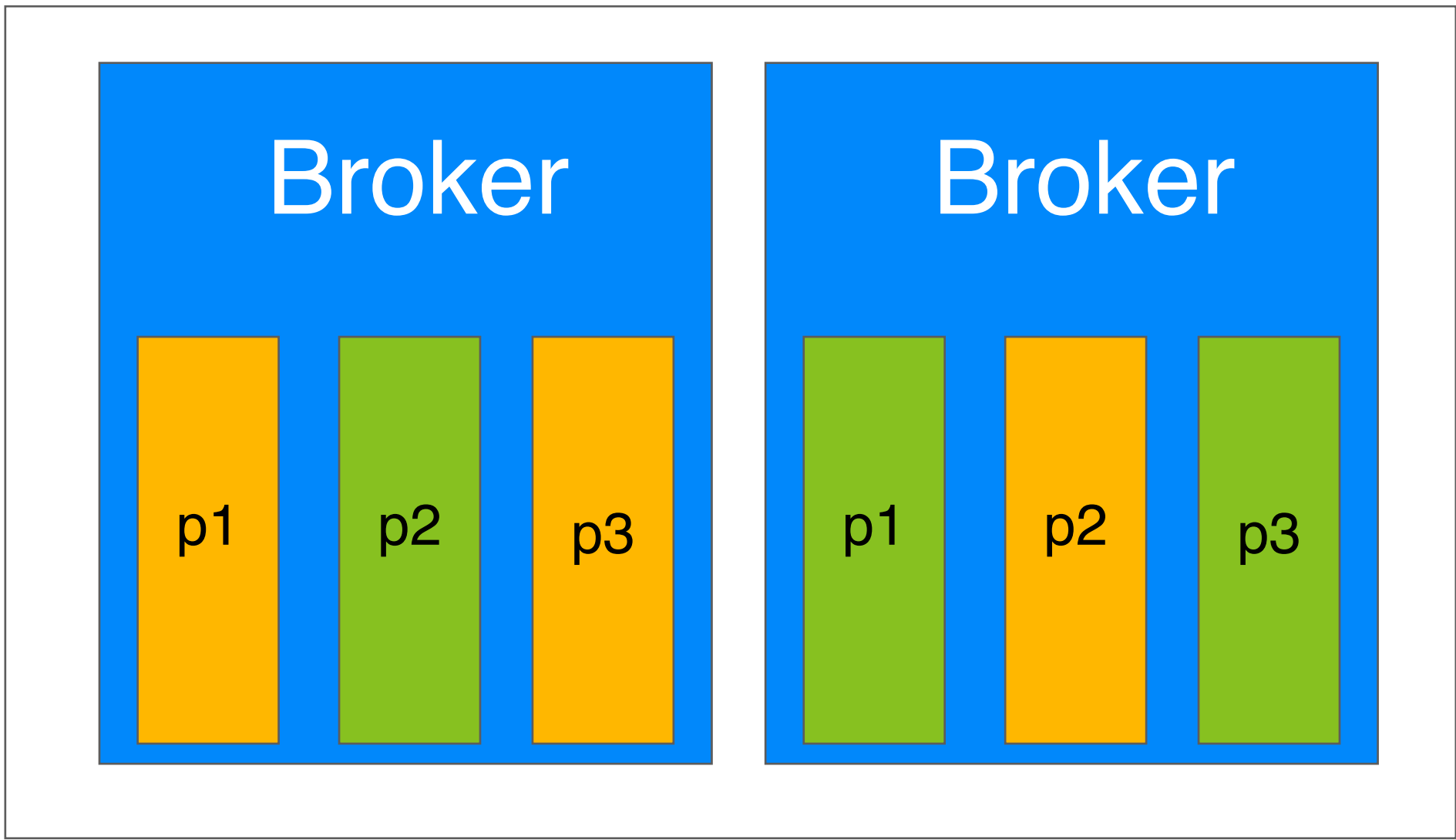
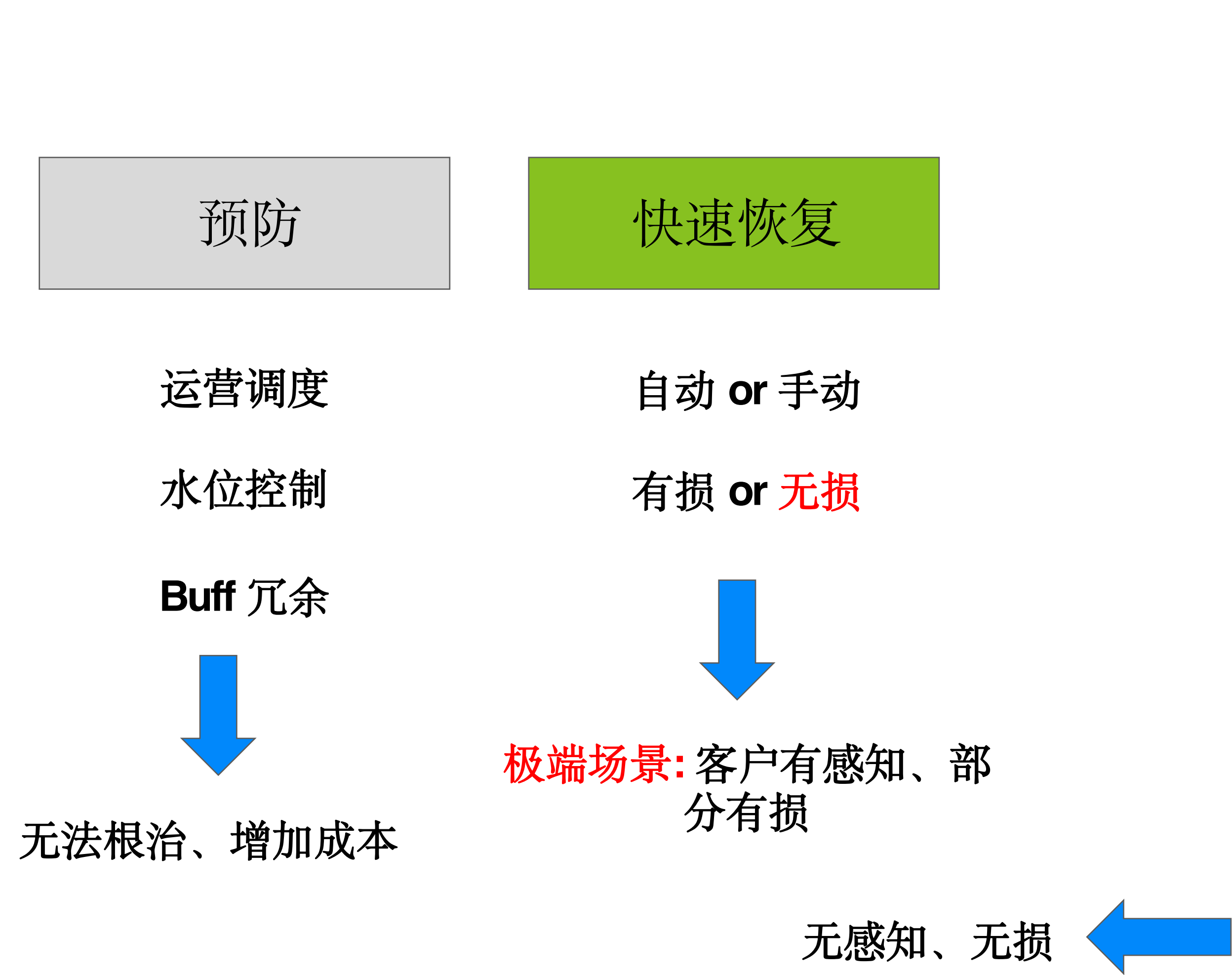
有状态Kafka服务计算层



1. 分区是消息数据保存的最小维度
2. 每个分区对应一个文件
3. 数据是分区有序的
4. **扩容需要迁移分区**

Kafka的架构不支持快速弹性

当前社区架构提升稳定性的思路



快速恢复:

1. 垂直升配节点规格 => leader切换
2. 定向扩容分区 => 消息乱序
3. 分区禁写 => 原生SDK不支持
4. 不迁移老数据 => 丢数据

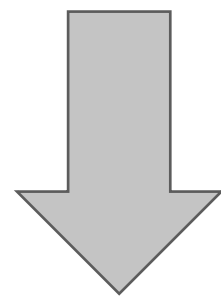
实现计算层快速弹性的方案

少迁移数据，不迁移数据

核心思路：有状态计算层 => 无状态计算层

1. 尽量迁移少量的老数据 => 【分层存储】

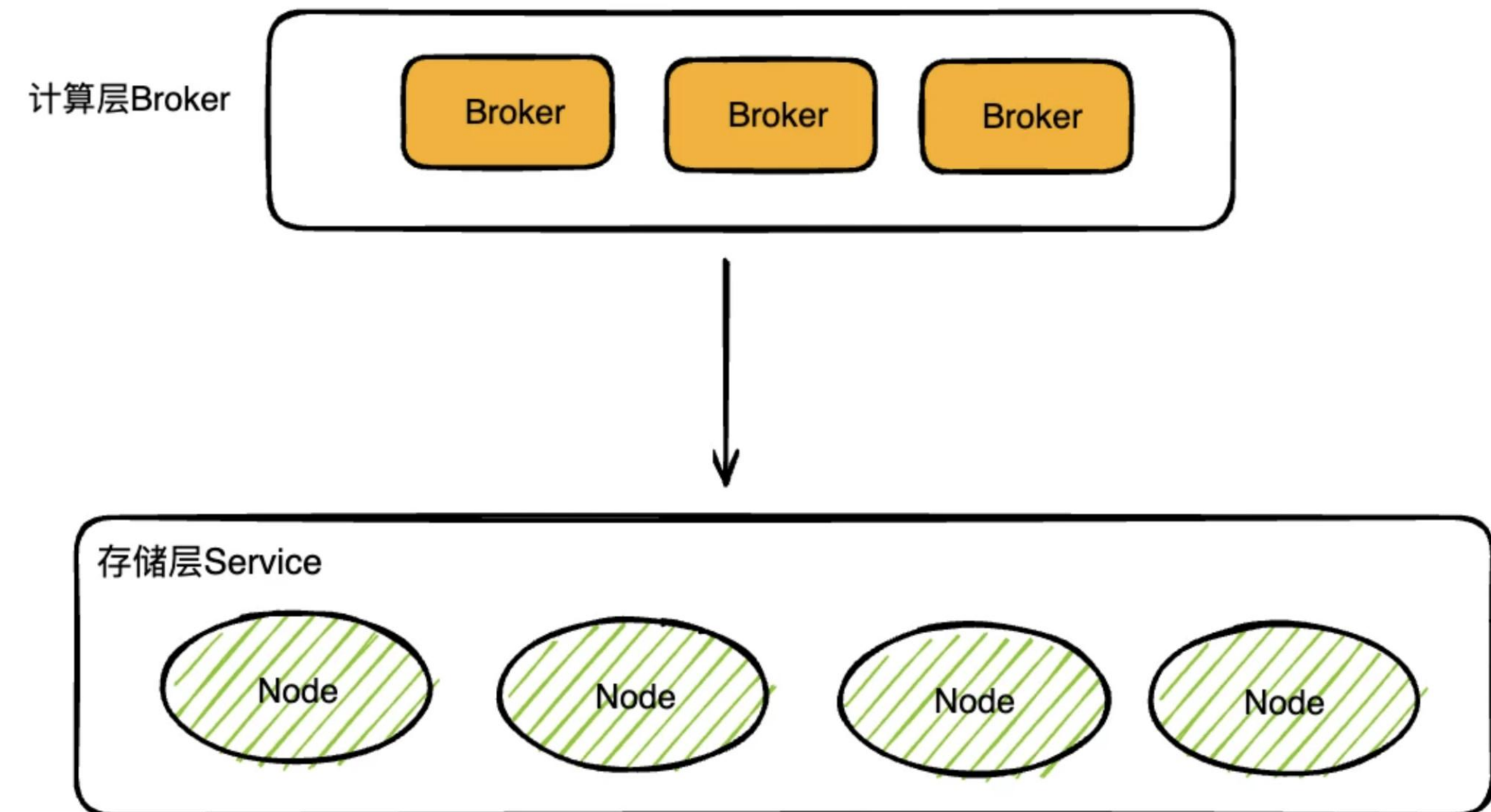
2. 计算存储分离，弹性的计算层



1. 云盘快速挂载新节点

2. 使用分布式文件系统存储数据

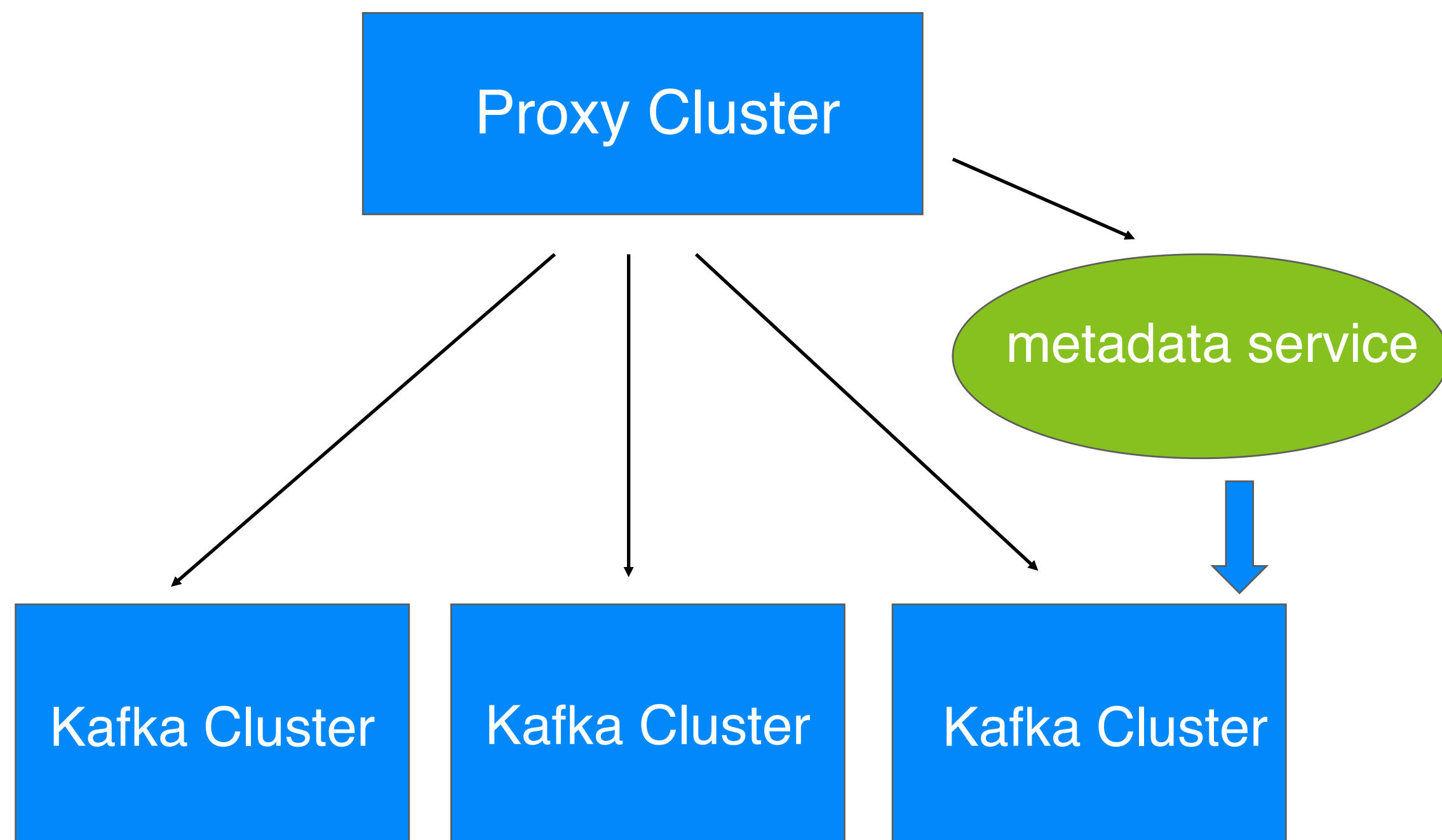
3. 计算存储分离架构



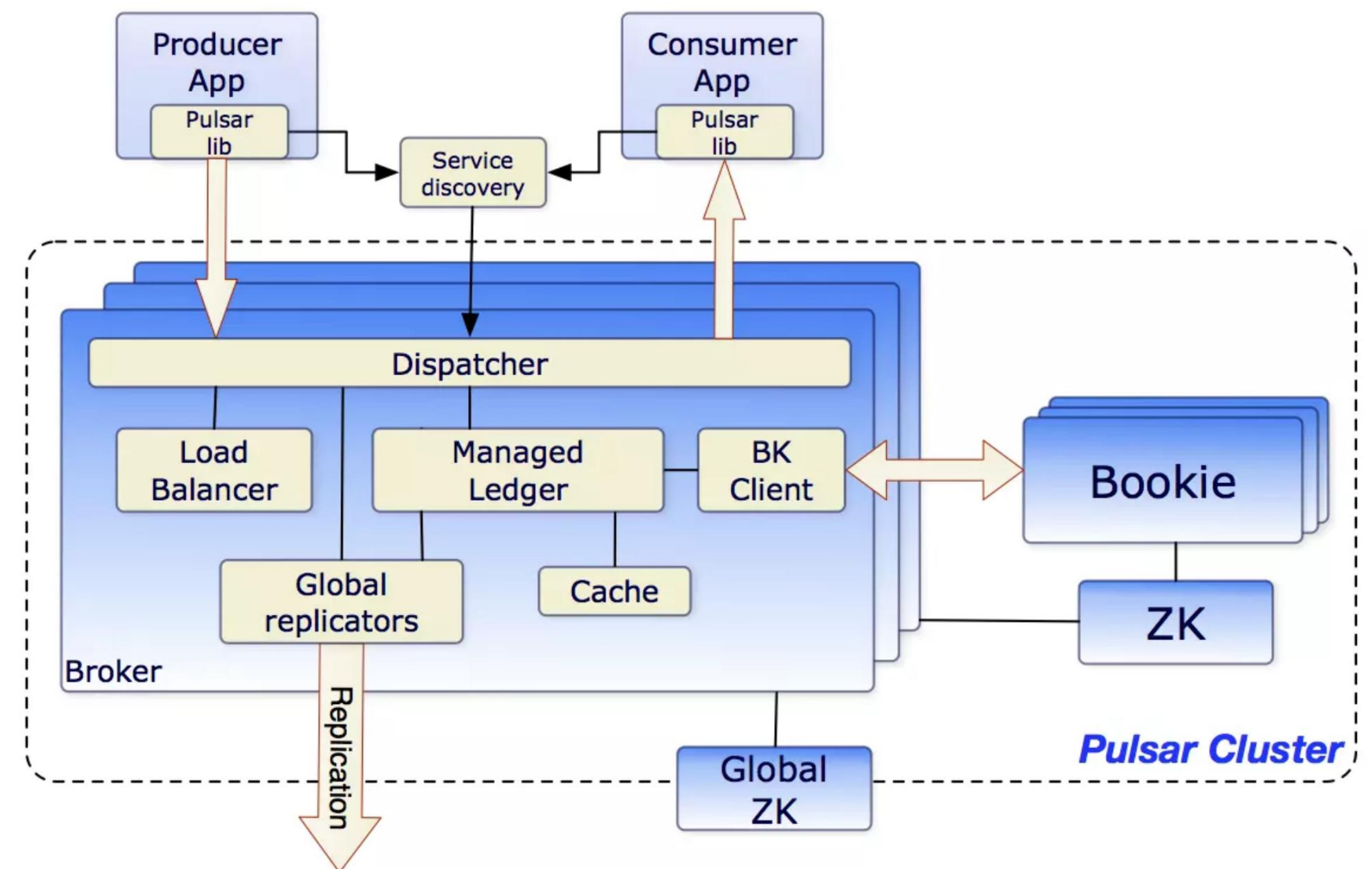
独立的计算层和存储层

计算存储分离架构

1. **Proxy**方案支持
2. 底层存储层切换远端引擎
3. 内核支持(架构改动太大、投入周期太长)



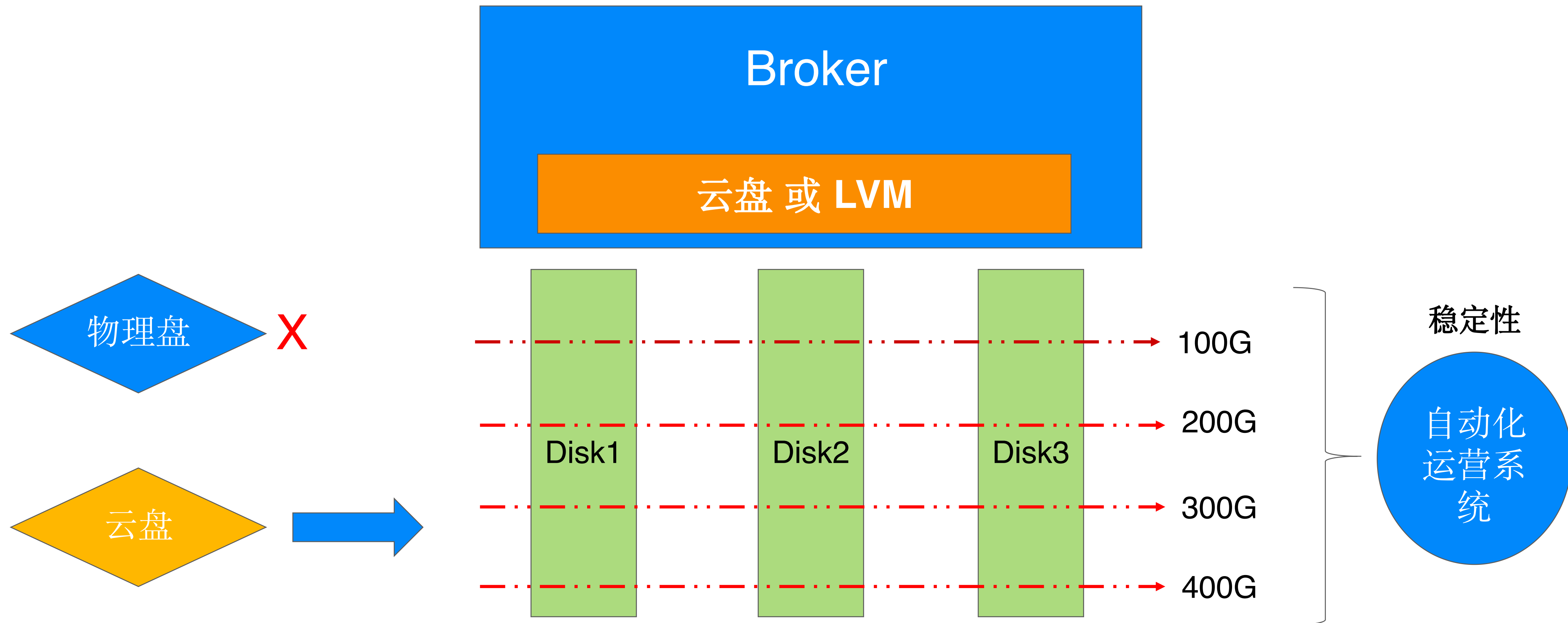
Pulsar 设计理念上是Kafka的升级版，核心解决弹性问题。



Pulsar的架构

弹性的本地存储架构

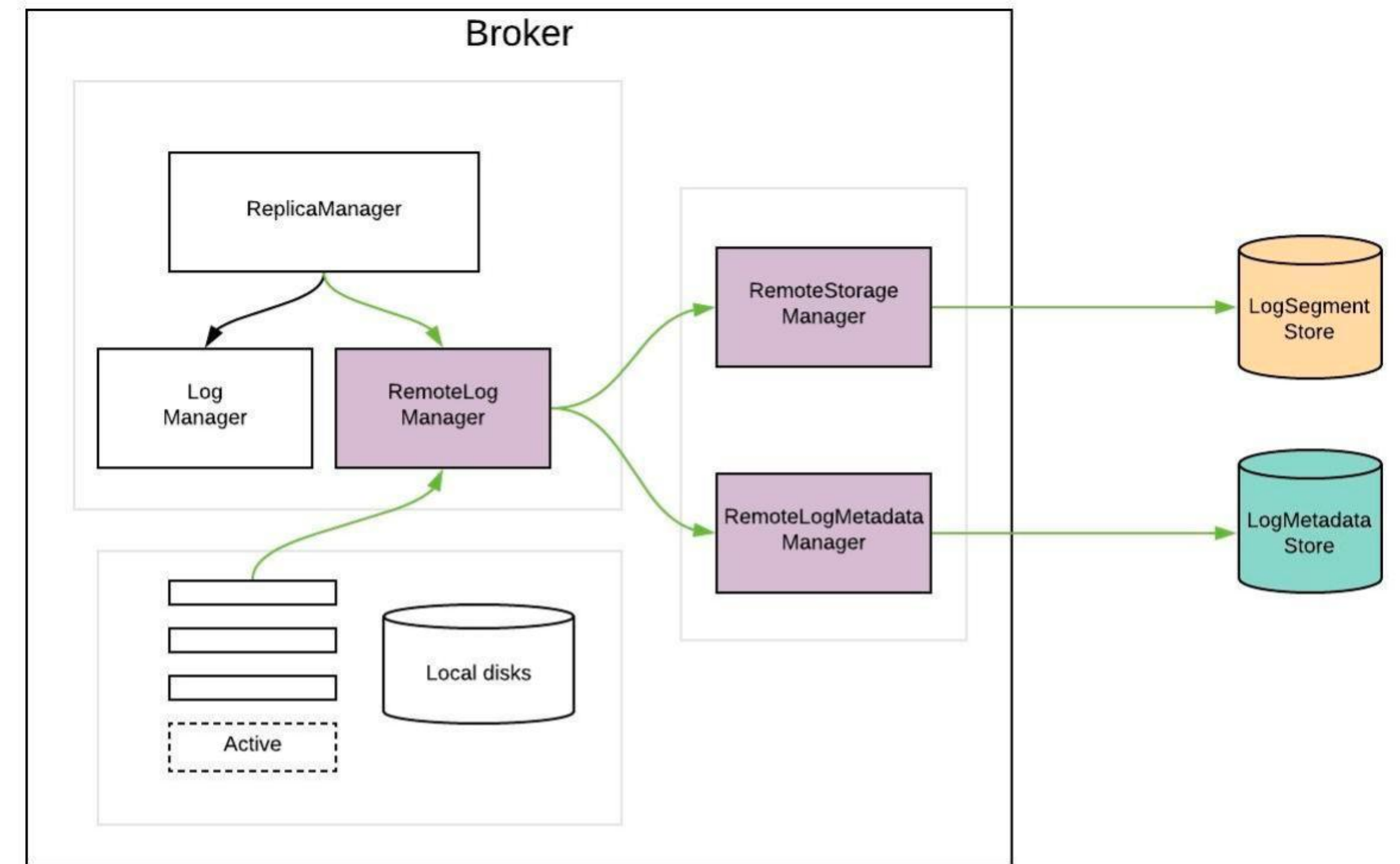
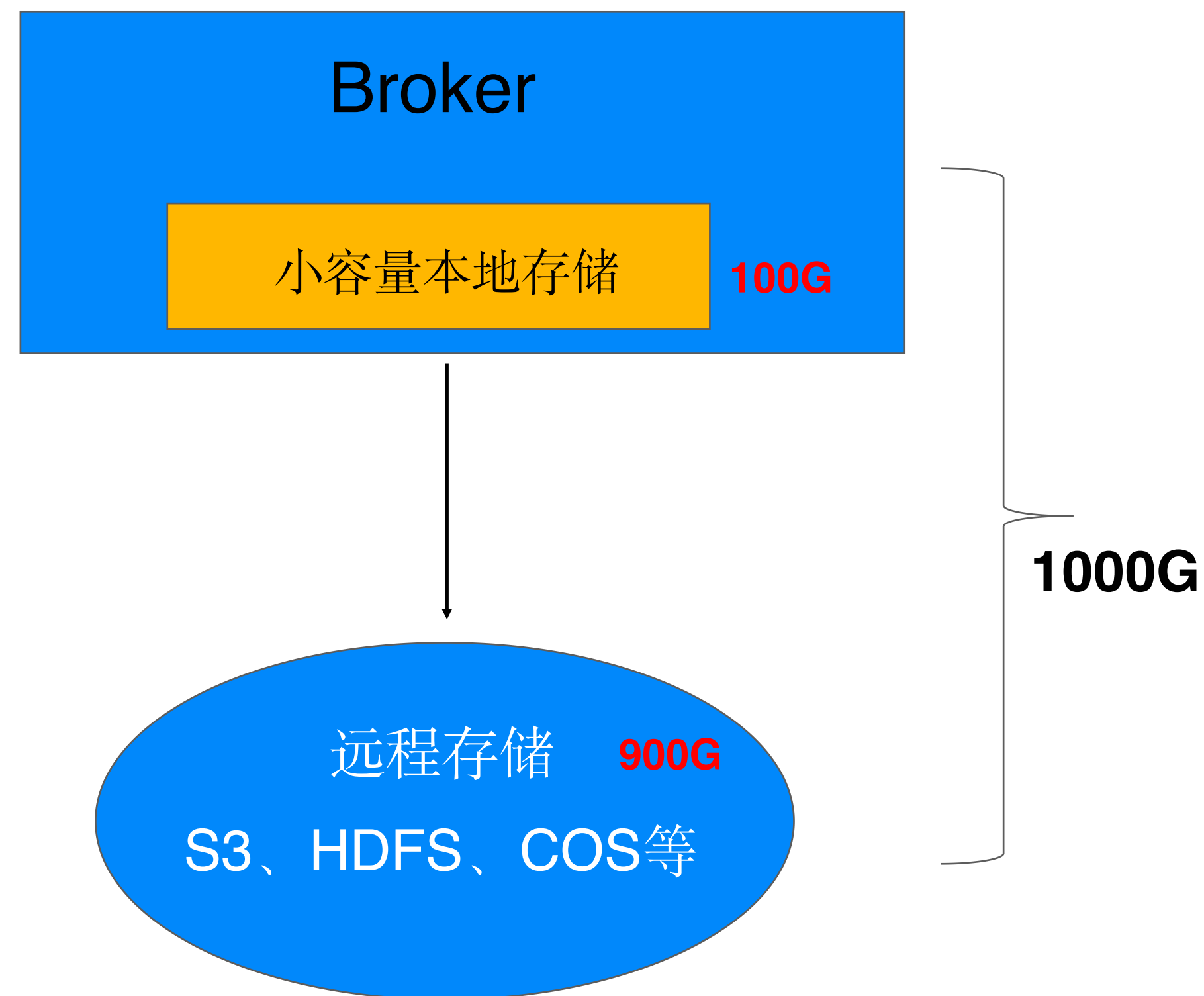
1. 核心是: 云盘 + 运营调度系统
2. 多块云盘 + 多目录结构 + 运营调度系统
3. 云盘 + LVM逻辑卷 + 运营调度系统



弹性的远程存储架构

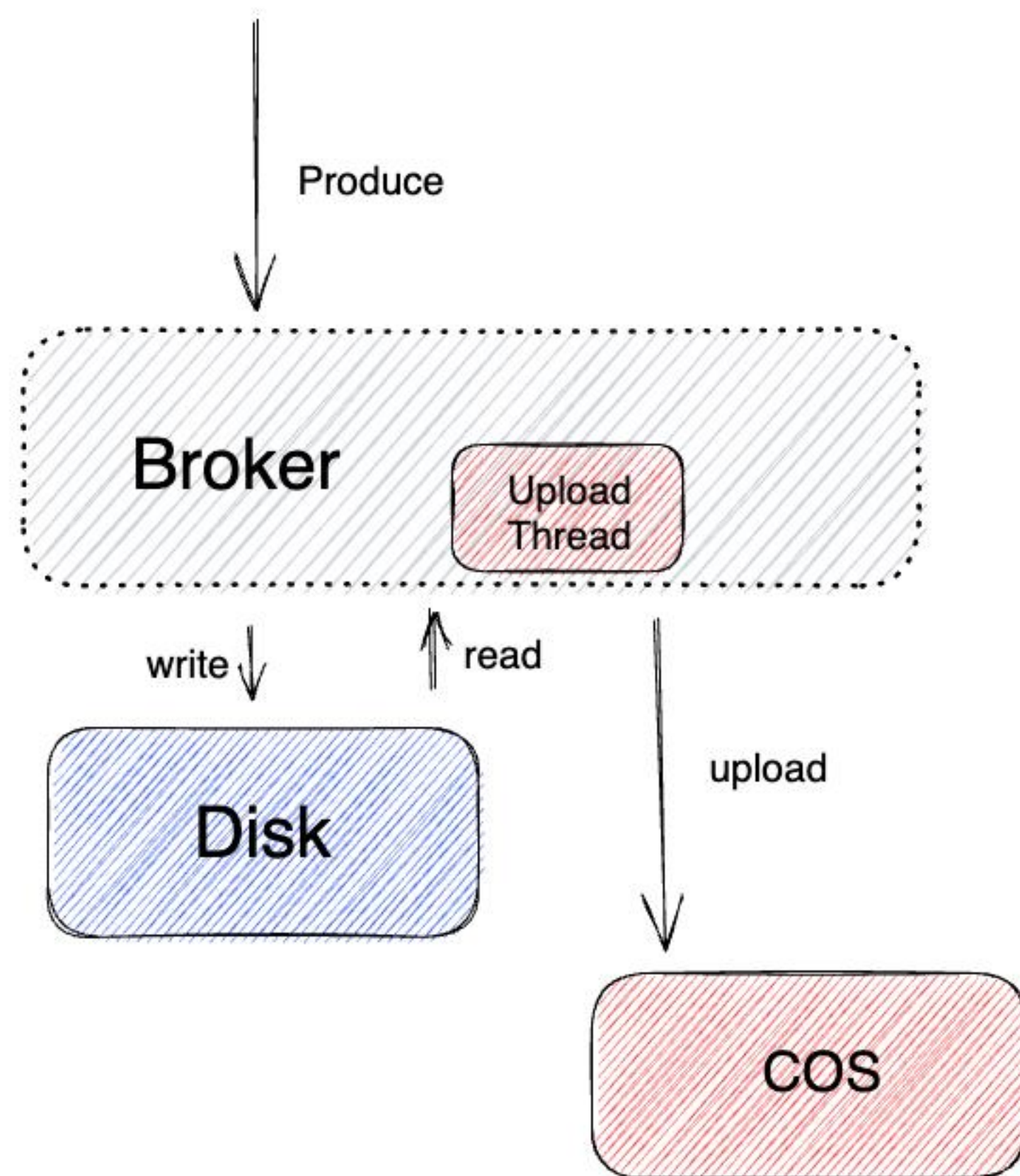
1. 通过更廉价的存储进一步降低成本

2. $1000 > 100 \times 1 + 900 \times 0.3 = 370$

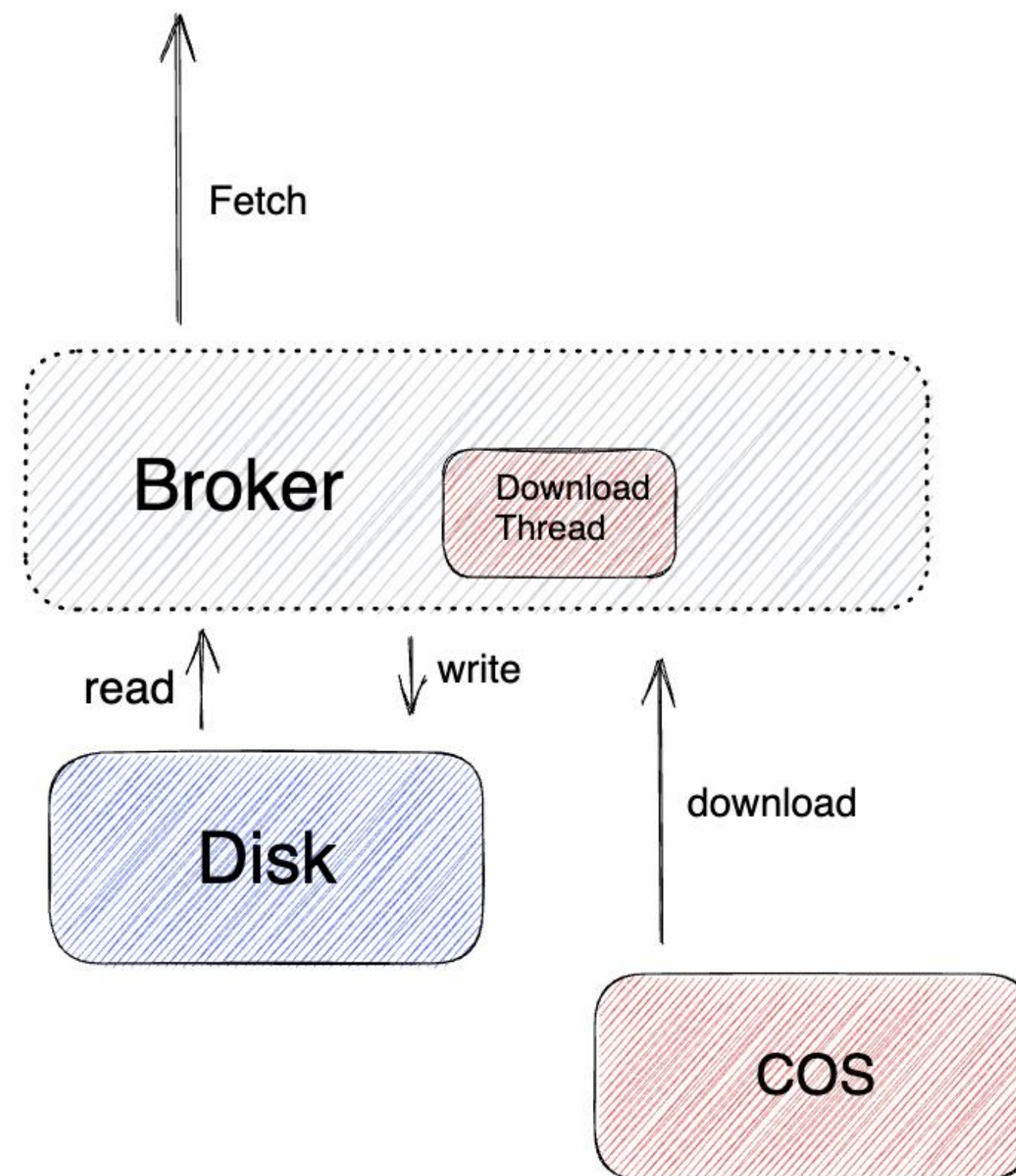


Kafka 分层存储

远程存储架构的稳定性



分层生产流程



分层消费流程

1. 硬盘: 独立的 IO 盘
2. 带宽: 上传下载流量和并行度限制
3. 内存: 尽量不使用内存缓存, 走硬盘缓存。细致的内存管理。
4. CPU: 线程隔离、线程绑核、限流
5. 回滚: 暂定分层的能力、止损和回滚预案。

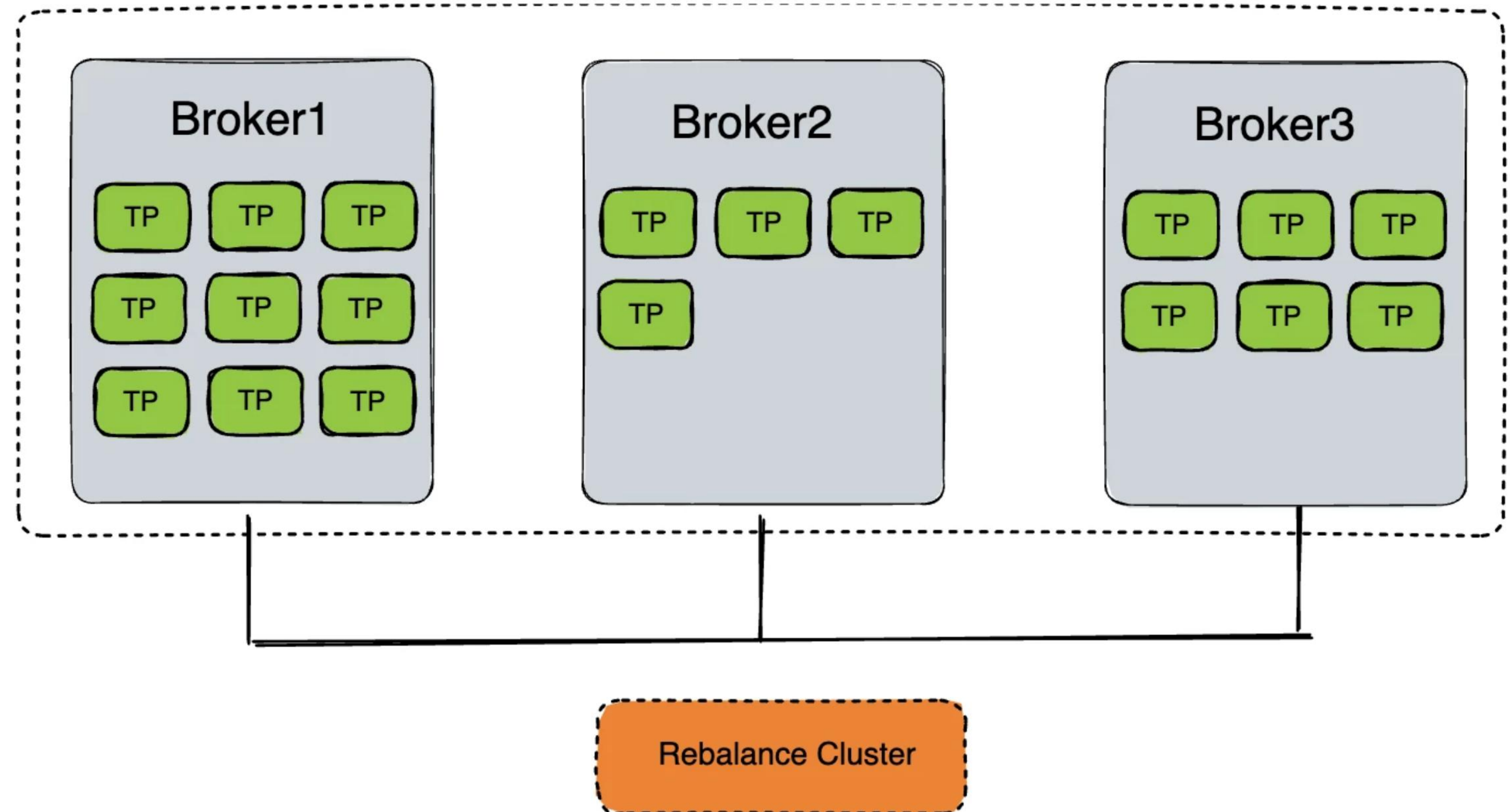
集群自均衡(Self Rebalance)

数据倾斜典型场景:

1. 节点间Topic和分区数量不均
2. 节点间流量不均
3. 节点间负载不均

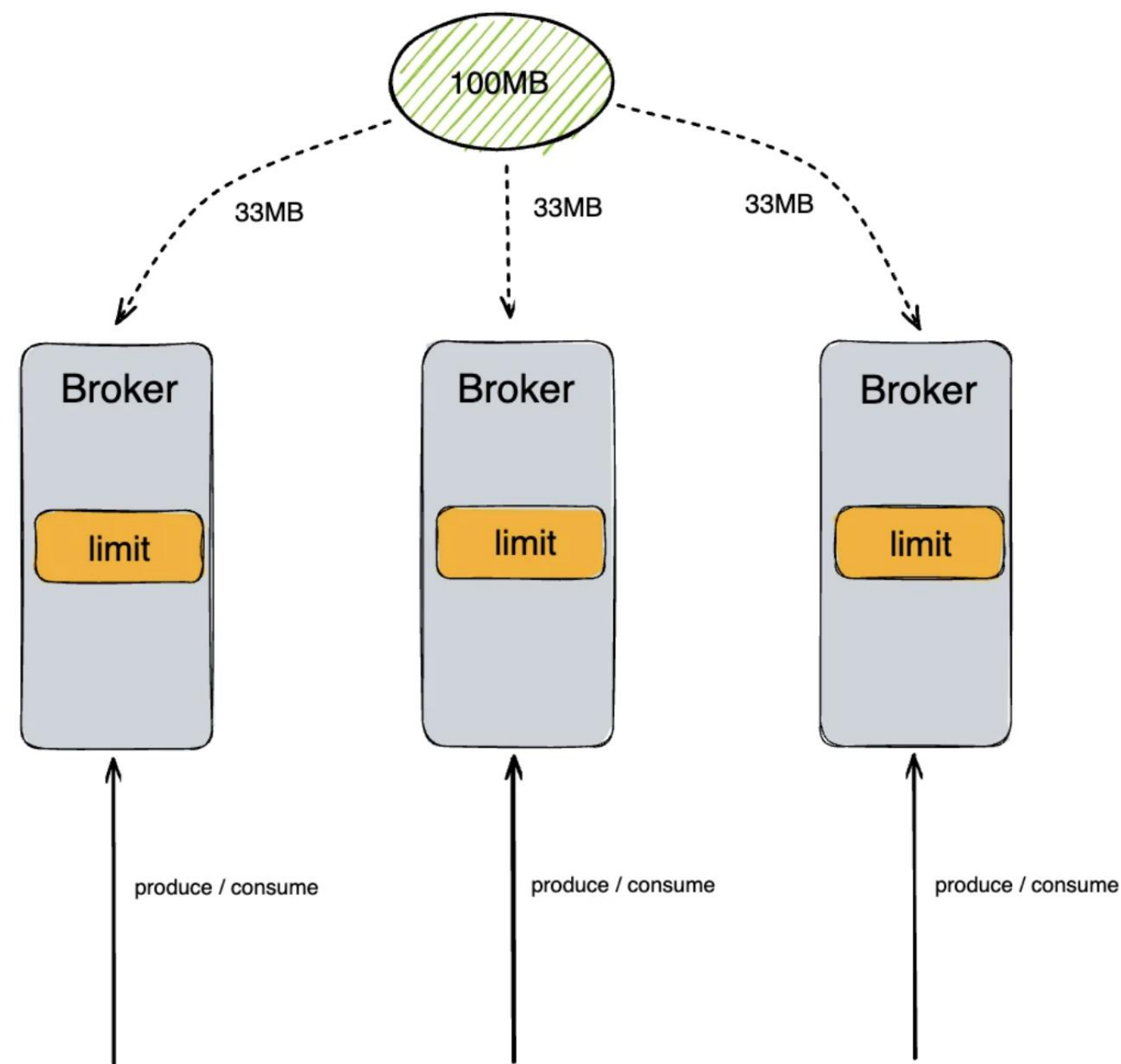
rebalance条件限制:

1. 校验Leader切换的影响
2. 低峰执行均衡
3. 均衡前的严格校验, 如流量、网络限制
4. 均衡对象的选择, 比如白名单、黑名单

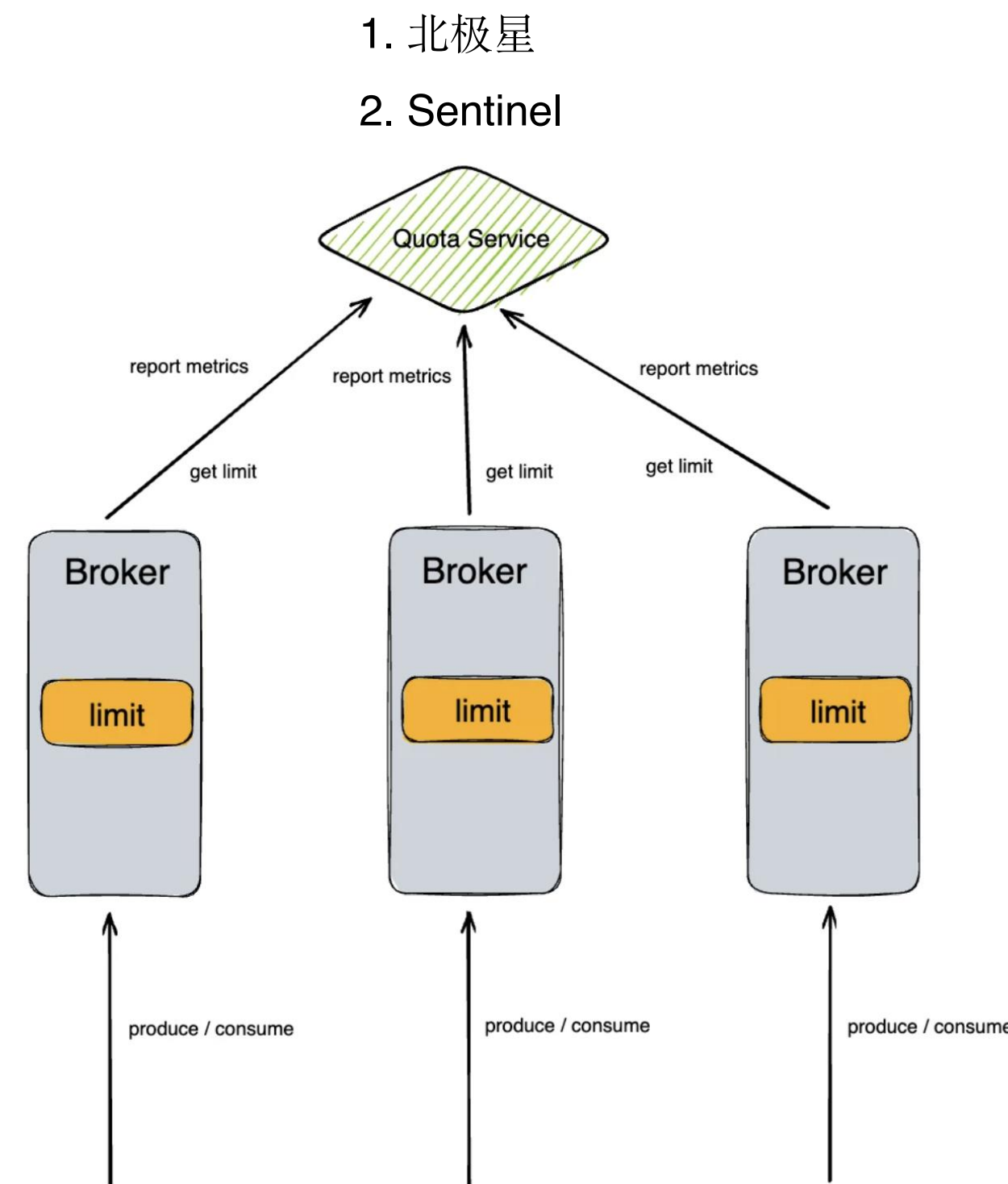


限流策略和机制的设计

1. 限流是集群自我保护的核心功能
2. 当前主要是单机模式限流，合理的应该是全局模式限流

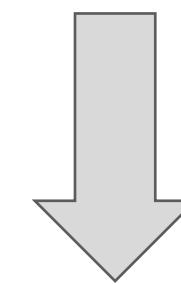


单机模式限流



全局模式限流

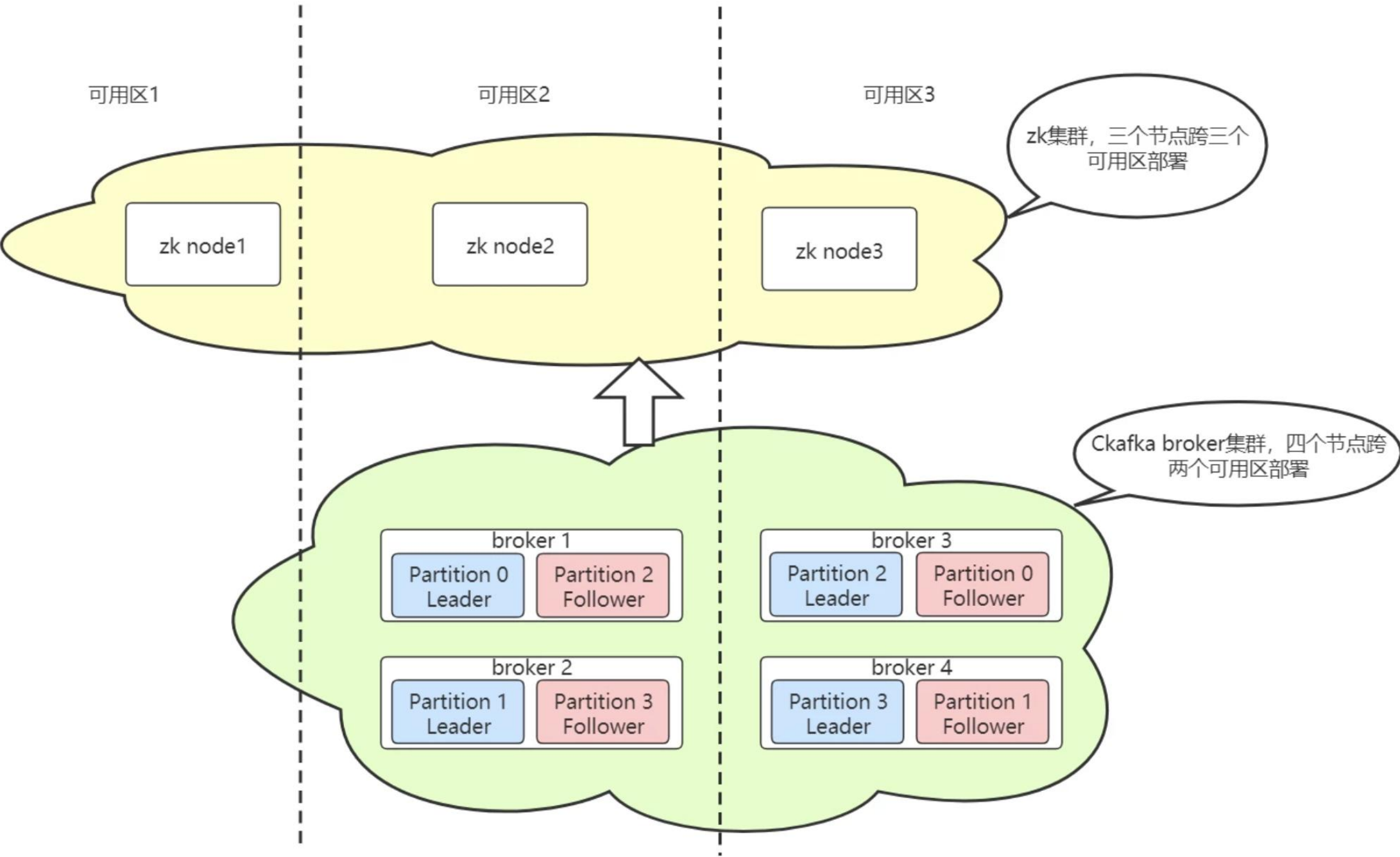
1. 是否会影响生产消费延时
2. 限流组件是否会成为瓶颈
3. 限流组件抖动如何处理



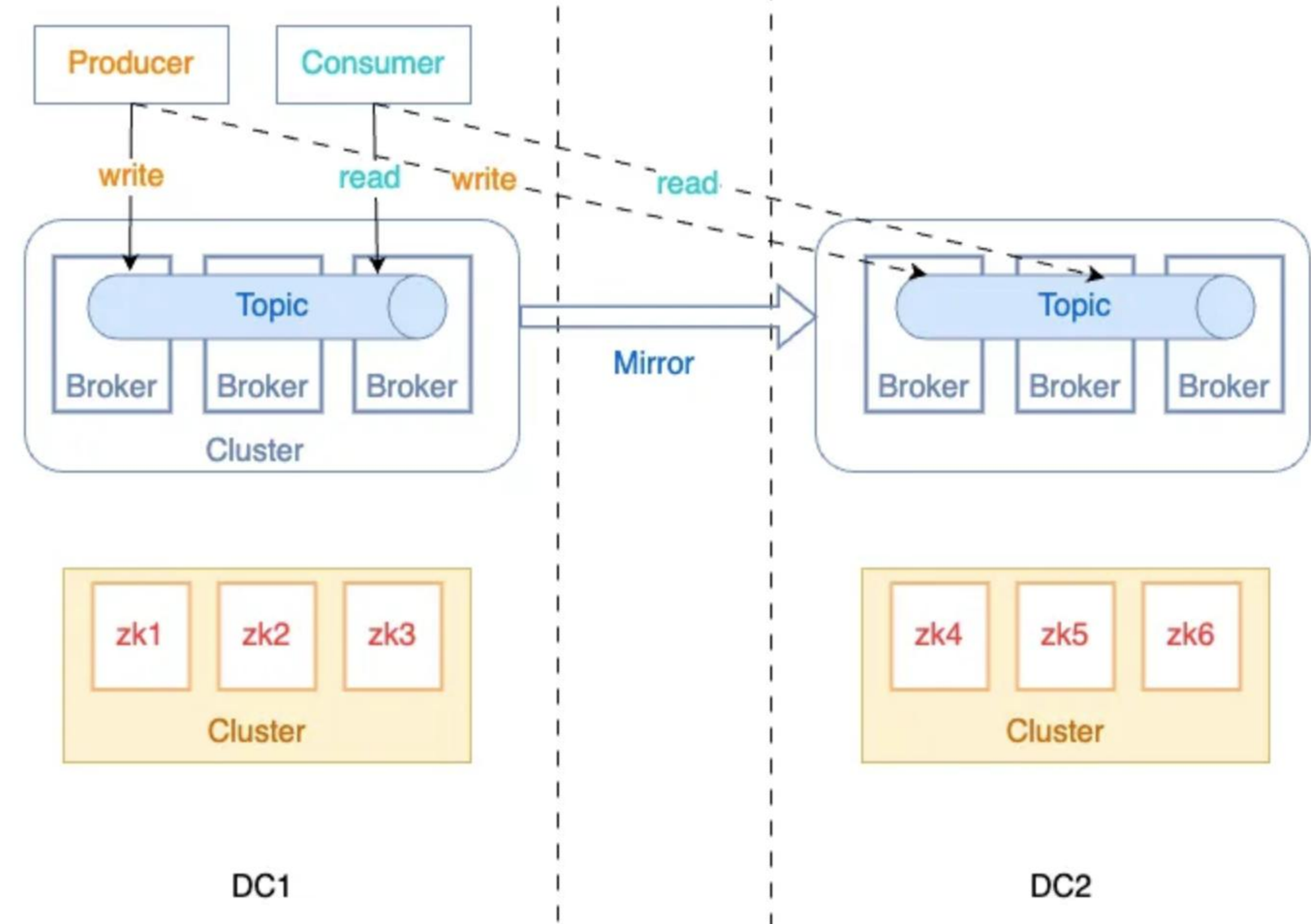
限流降级策略

跨区、跨地域容灾能力建设

控制副本分布、跨集群数据复制



跨区容灾架构



跨地域容灾架构

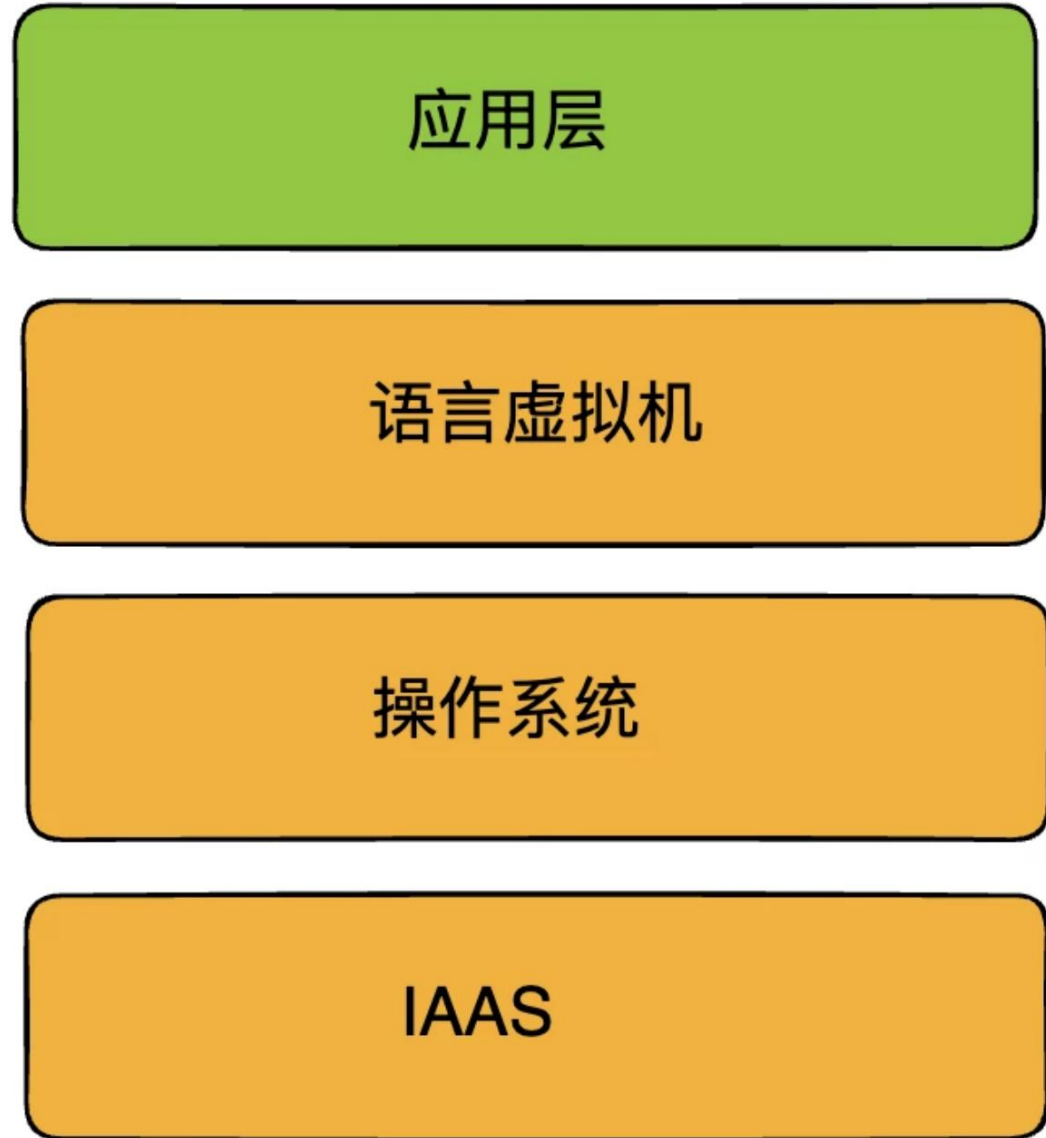
混沌工程

在云上大规模Kafka集群的落地实践

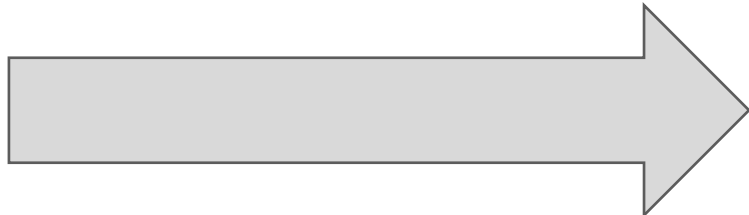
1. 演练场景梳理
2. 日常演习和故障模拟
3. 快速复制现网流量
4. 演练风险控制

Kafka 的演练场景梳理

不同架构层的场景是不一样的，主要关注应用层。



单机四层架构



演练场景梳理

网络	ip封堵
功能	路由策略添加异常
	路由策略删除
	ACL策略读写拒绝
	ACL策略删除

容器	broker pod重建
	node 封锁
	pod 配置更新
进程	broker节点重启
	zookeeper节点重启
	broker cpu 高负载
	broker 磁盘io消耗
	broker 进程挂起
	broker 进程kill
	broker 内存高负载
	zookeeper node节点删除
	单broker数据文件删除

类别	场景
主机	单broker宕机
	双broker宕机
	单zookeeper宕机
	双zookeeper宕机
	broker节点扩容
	broker节点缩容
	系统时间跳变
	iptables drop
可用区	单az宕机
	多az主备切换

日常演习和故障模拟

演习分类	产品模块	演习类型	演习项目（演习内容和目的）	预期演习时间 （仅研发识别的 演习场景填写）	演习频率
主机 可用区	broker	蓝军演习	容灾架构高可用规划场景	/	每月
主机 可用区	zk	蓝军演习	容灾架构高可用规划场景	/	每月
主机 可用区	控制流	蓝军演习	容灾架构高可用规划场景	/	每月
容器 主机 进程 可用区 网络	broker	常规演习	架构高可用，单点故障 监控告警有效性 特定配置、参数验证	/	每月
主机 进程 可用区 网络	zk	常规演习	架构高可用，单点故障 监控告警有效性 特定配置、参数验证	/	每月
容器 主机 进程 可用区 网络	控制流	常规演习	架构高可用，单点故障 监控告警有效性 特定配置、参数验证	/	每月

通用场景模拟：

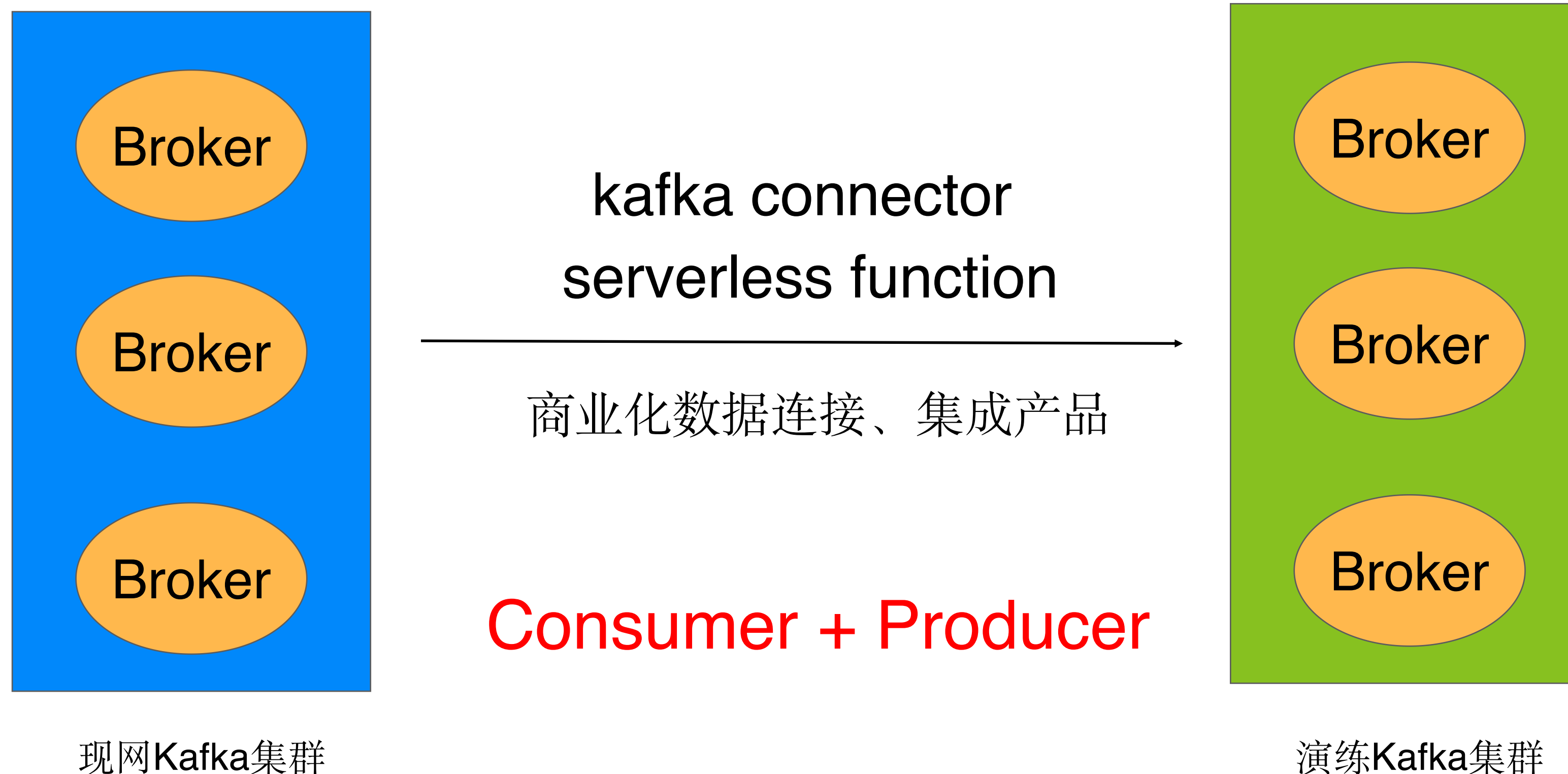
- 1. 内部混沌平台
- 2. 开源工具，如ChaosBlade、Chaos Mesh
- 3. 简单工具和自建工具

业务场景模拟

节点间 Topic和分区数量不均	自动化运营调度系统
Topic 分区数流量不均	
Topic 分区数太少	
集群元数据不一致	
负载类	现网流量模拟

如何稳定复制现网流量

MQ Connector、Flink、Serverless Function、数据集成套件等等



演练本身的稳定性风险控制

- 1. 根据不同版本，不同架构搭建对应的演练环境
- 2. 定义高、中、低风险的演练场景。
- 3. 控制人工、自动化的演练比例。
- 4. 重启类操作的API限制或不提供API，尽量人工介入。

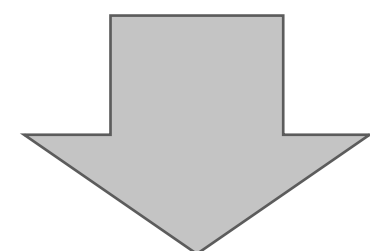
	线下环境	预发布环境	线上环境
可控性	高	较高	低
演习成本	中	较高	高
人力成本	低	中	高
频率	高	中	低

云上消息队列在 降本、稳定性、架构升级之间权衡的一些思考

稳定性优先下的成本优化策略

稳定性 > 成本

1. 集群粒度差异化的成本优化策略。
2. 更细致的运营调度分析体系，让时间差带来的价值。
3. 架构升级提升软件性能、按量付费、丰富产品特性 => 置换成本优化后SLA



双赢

未来展望

1. 完善分层存储架构。在不大规模改动内核的情况下，降低扩缩容的成本。
2. Kafka 在存算分离方向的探索。
3. 故障自治、自愈的集群能力建设。
4. 围绕稳定性、故障处理、成本优化的自动化运维运营体系建设。

JUST GIVE IT A TRY

**Faster data. Lower bills.
More time for coffee.**

Redpanda is a Kafka®-compatible streaming data platform that is proven to be 10x faster, and 6x lower in total costs for GBps+ throughputs. It is also JVM-free, ZooKeeper®-free, Jepsen-tested and source available.

BENCHMARKS

DOCUMENTATION



redpanda

我希望有一个： 稳定的、**serverless**的、支持多协议、架构简单的消息队列。

用 **Rust** 写一个存算分离、兼容多协议的 **MQ** 【正在做, 个人兴趣~】

今年六月中旬上线(下个月)



扫一扫上面的二维码图案，加我为朋友。



消息队列核心技术 40 讲

全面掌握消息队列架构原理和设计思路

许文强

腾讯云 Kafka 技术负责人



你将获得

- ✓ 1 张 MQ 开发必备知识地图
- ✓ 4 款主流 MQ 通用原理解读
- ✓ 最简代码体验 MQ 功能设计思路
- ✓ 以 Pulsar 为代表的最新 MQ 探索



扫码下载极客时间 App

THANKS



软件正在重新定义世界

Software Is Redefining The World